

4

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

HIROAKI KIMURA et al.

Serial No. 09/241,735

Filed: February 2, 1999

For: An Apparatus for Analyzing
Software and Method of the Same

Atty Docket: 1776/00034



**SUBMISSION OF CERTIFIED PRIORITY DOCUMENT(S) and
CLAIM TO PRIORITY UNDER 35 U.S.C. § 119**

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Priority under 35 U.S.C. § 119 is hereby claimed to the following priority documents, a certified copy of which is enclosed. The document was filed in a foreign country within the proper statutory period prior to the filing of the above-referenced United States patent application.

<u>Country</u>	<u>Priority Document Serial No.</u>	<u>Filing Date</u>
Japan	10-024357	February 5, 1998

Acknowledgement of this claim and submission in the next official communication is respectfully requested.

Respectfully submitted,

for Elzbieta Chlopecka 42,709
ELZBIETA CHLOPECKA
Registration No. 32,767
Pollock, Vande Sande & Amernick, R.L.L.P.
1990 M Street, N.W.
Washington, D. C. 20036-3425
Telephone: 202-331-7111

Date: 4/21/99

日本国特許庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
in this Office.

願年月日
Date of Application:



願番号
Application Number:

平成10年特許願第024357号

願人
Applicant(s):

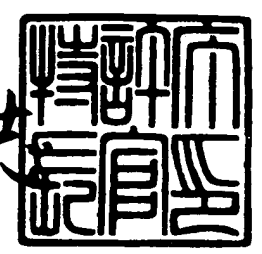
新日本製鐵株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

1999年 2月26日

特許庁長官
Commissioner,
Patent Office

伴佐山建志



【書類名】 特許願

【整理番号】 SINP1093

【提出日】 平成10年 2月 5日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/00
G06F 11/00

【発明の名称】 ソフトウェア解析装置及びソフトウェア解析方法

【請求項の数】 24

【発明者】

 【住所又は居所】 東京都千代田区大手町 2 丁目 6 番 3 号 新日本製鐵株式
会社内

 【氏名】 木村 博昭

【発明者】

 【住所又は居所】 東京都千代田区大手町 2 丁目 6 番 3 号 新日本製鐵株式
会社内

 【氏名】 野口 正浩

【特許出願人】

 【識別番号】 000006655

 【氏名又は名称】 新日本製鐵株式会社

 【代表者】 浅村 峻

【代理人】

 【識別番号】 100091269

 【弁理士】

 【氏名又は名称】 半田 昌男

【手数料の表示】

 【予納台帳番号】 007571

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

特平 10-024357

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9710440

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ソフトウェア解析装置及びソフトウェア解析方法

【特許請求の範囲】

【請求項1】 コンピュータプログラムのソースコードを読み込んで、自動的にプログラム解析情報を作成するプログラム解析情報作成手段と、

該プログラム解析情報作成手段によって前記プログラム解析情報が作成され次第、適当な単位で分割して所定のデータ記録媒体に格納するプログラム解析情報格納手段と、

前記データ記録媒体から前記プログラム解析情報を読み出して、プログラム解析を実行するプログラム解析手段と、

を有することを特徴とするソフトウェア解析装置。

【請求項2】 前記プログラム解析手段は、前記データ記録媒体から前記プログラム解析情報を読み出して、オペレータとの間の対話的処理によりプログラム解析を実行することを特徴とする請求項1記載のソフトウェア解析装置。

【請求項3】 前記プログラム解析情報格納手段は、前記プログラム解析情報をデータベース化して前記データ記録媒体に格納することを特徴とする請求項1又は2記載のソフトウェア解析装置。

【請求項4】 前記データ記録媒体から前記プログラム解析情報を読み出して、バッチ解析処理により、メトリクス情報、冗長度情報、データフロー異常情報、保守用文書情報のうち少なくとも一つを作成することを特徴とする請求項1、2又は3記載のソフトウェア解析装置。

【請求項5】 前記プログラム解析情報には、
前記ソースコードに基づいて作成される構文解析木、
前記ソースコードに基づいて前記ソースコードに用いられているシンボルの意味が記述されたシンボルテーブル、

前記構文解析木に基づいて作成されるコールグラフあるいはフローグラフ、
前記構文解析木、シンボルテーブル、フローグラフ、コールグラフに基づいて作成されるデータフロー情報、

前記構文解析木、シンボルテーブル、コールグラフ、フローグラフ、データフ

ロー情報に基づいて作成されるプログラム依存グラフ、

およびモジュール入出力情報、

のうち少なくとも一つが含まれていることを特徴とする請求項1, 2, 3又は4記載のソフトウェア解析装置。

【請求項6】 前記プログラム解析情報作成手段は、前記構文解析木及びシンボルテーブル、コールグラフ及びフローグラフ、データフロー情報、並びにプログラム依存グラフ及びモジュール入出力情報を、この順序で作成し、出力することを特徴とする請求項1, 2, 3, 4又は5記載のソフトウェア解析装置。

【請求項7】 コンピュータプログラムのソースコードからプログラムの解析に必要な情報を抽出し、抽出された前記情報を解析する目的別に階層的にデータベースに登録し、

プログラムの解析を行うときは解析する目的に応じて所定の階層に登録されている前記情報を読み出して解析を実行することを特徴とするソフトウェア解析装置。

【請求項8】 前記データベースは、オブジェクト指向型のデータベースであることを特徴とする請求項7記載のソフトウェア解析装置。

【請求項9】 コンピュータプログラムのソースコードを読み込んで、自動的にプログラム解析情報を作成するプログラム解析情報作成工程と、

該プログラム解析情報作成工程によって前記プログラム解析情報が作成され次第、適当な単位で分割して所定のデータ記録媒体に格納するプログラム解析情報格納工程と、

前記データ記録媒体から前記プログラム解析情報を読み出して、プログラム解析を実行するプログラム解析工程と、

を有することを特徴とするソフトウェア解析方向。

【請求項10】 前記プログラム解析工程は、前記データ記録媒体から前記プログラム解析情報を読み出して、オペレータとの間の対話的処理によりプログラム解析を実行することを特徴とする請求項9記載のソフトウェア解析方法。

【請求項11】 前記プログラム解析情報格納工程は、前記プログラム解析情報をデータベース化して前記データ記録媒体に格納することを特徴とする請求

項9又は10記載のソフトウェア解析方法。

【請求項12】 前記データ記録媒体から前記プログラム解析情報を読み出して、バッチ解析処理により、メトリクス情報、冗長度情報、データフロー異常情報、保守用文書情報のうち少なくとも一つを作成することを特徴とする請求項9、10又は11記載のソフトウェア解析方法。

【請求項13】 前記プログラム解析情報には、
前記ソースコードに基づいて作成される構文解析木、
前記ソースコードに基づいて前記ソースコードに用いられているシンボルの意味が記述されたシンボルテーブル、
前記構文解析木に基づいて作成されるコールグラフあるいはフローグラフ、
前記構文解析木、シンボルテーブル、フローグラフ、コールグラフに基づいて作成されるデータフロー情報、
前記構文解析木、シンボルテーブル、コールグラフ、フローグラフ、データフロー情報に基づいて作成されるプログラム依存グラフ、
およびモジュール入出力情報、
のうち少なくとも一つが含まれていることを特徴とする請求項9、10、11又は12記載のソフトウェア解析方法。

【請求項14】 前記プログラム解析情報作成手段は、前記構文解析木及びシンボルテーブル、コールグラフ及びフローグラフ、データフロー情報、並びにプログラム依存グラフ及びモジュール入出力情報を、この順序で作成し、出力することを特徴とする請求項9、10、11、12又は13記載のソフトウェア解析方法。

【請求項15】 コンピュータプログラムのソースコードからプログラムの解析に必要な情報を抽出し、抽出された前記情報を解析する目的別に階層的にデータベースに登録し、

プログラムの解析を行うときは解析する目的に応じて所定の階層に登録されている前記情報を読み出して解析を実行することを特徴とするソフトウェア解析方法。

【請求項16】 前記データベースは、オブジェクト指向型のデータベース

であることを特徴とする請求項 15 記載のソフトウェア解析方法。

【請求項 17】 コンピュータに、

コンピュータプログラムのソースコードを読み込んで、自動的にプログラム解析情報を作成するプログラム解析情報作成機能と、

該プログラム解析情報作成機能によって前記プログラム解析情報が作成され次第、適当な単位で分割して所定のデータ記録媒体に格納するプログラム解析情報格納機能と、

前記データ記録媒体から前記プログラム解析情報を読み出して、プログラム解析を実行するプログラム解析機能と、

を実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 18】 前記プログラム解析機能は、前記データ記録媒体から前記プログラム解析情報を読み出して、オペレータとの間の対話的処理によりプログラム解析を実行することを特徴とする請求項 17 記載のコンピュータ読み取り可能な記録媒体。

【請求項 19】 前記プログラム解析情報格納機能は、前記プログラム解析情報をデータベース化して前記データ記録媒体に格納することを特徴とする請求項 17 又は 18 記載のプログラム読み取り可能な記録媒体。

【請求項 20】 前記データ記録媒体から前記プログラム解析情報を読み出して、バッチ解析処理により、メトリクス情報、冗長度情報、データフロー異常情報、保守用文書情報のうち少なくとも一つを作成する機能を実行させるコンピュータプログラムが記録されていることを特徴とする請求項 17, 18 又は 19 記載のコンピュータ読み取り可能な記録媒体。

【請求項 21】 前記プログラム解析情報には、

前記ソースコードに基づいて作成される構文解析木、

前記ソースコードに基づいて前記ソースコードに用いられているシンボルの意味が記述されたシンボルテーブル、

前記構文解析木に基づいて作成されるコールグラフあるいはフローグラフ、

前記構文解析木、シンボルテーブル、フローグラフ、コールグラフに基づいて作成されるデータフロー情報、

前記構文解析木、シンボルテーブル、コールグラフ、フローグラフ、データフロー情報に基づいて作成されるプログラム依存グラフ、

およびモジュール入出力情報、

のうち少なくとも一つが含まれていることを特徴とする請求項 17, 18, 19 又は 20 記載のコンピュータ読み取り可能な記録媒体。

【請求項 22】 前記プログラム解析情報作成機能は、前記構文解析木及びシンボルテーブル、コールグラフ及びフローグラフ、データフロー情報、並びにプログラム依存グラフ及びモジュール入出力情報を、この順序で作成し、出力することを特徴とする請求項 17, 18, 19, 20 又は 21 記載のコンピュータ読み取り可能な記録媒体。

【請求項 23】 コンピュータプログラムのソースコードからプログラムの解析に必要な情報を抽出し、抽出された前記情報を解析する目的別に階層的にデータベースに登録し、

プログラムの解析を行うときは解析する目的に応じて所定の階層に登録されている前記情報を読み出して解析を実行する機能を実行するコンピュータプログラムが記録されていることを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 24】 前記データベースは、オブジェクト指向型のデータベースであることを特徴とする請求項 23 記載のコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、コンピュータプログラムの内容を理解し易くするために、コンピュータプログラムについて解析を行うソフトウェア解析装置及びソフトウェア解析方法に関連する。

【0002】

【従来の技術】

多数のプログラマが協力して大規模なプログラムを開発する場合や、既にあるプログラムの保守を行おうとする場合、他人が書いたプログラムを理解しなけれ

ばならないことがある。しかし、他人が書いたプログラムを理解することは、大きな困難を伴う。

【0003】

そこで、プログラムのソースコードを自動的に解析して様々な情報を提供することにより、プログラムの理解を支援する装置が提案されている。その一例としてコールグラフやフローグラフを作成し、プログラムを視覚的に把握できるようにする装置及び方法がある。これについては、例えば、本出願人による特願平9-32415、特願平9-32452などを参照することができる。

【0004】

【発明が解決しようとする課題】

しかしながら、大規模なプログラム、例えば100万ステップを超えるようなプログラムに対して解析を行おうとすると、解析情報のボリュームも非常に大きくなる。このため、解析情報が、通常のワークステーションやパーソナルコンピュータが実装できる最大容量のメモリに入りきらないとか、解析に要する時間が非常に長くかかるという状況が起こる。例えば、解析途中でメモリ容量が不足し、解析情報が入らなくなってしまった場合には、解析はそれ以上先に進めることはできないので、何らかの対応策を講じた上で、最初からプログラムの解析を再度行わなければならない、作業の効率を悪くしていた。

【0005】

本発明は、上記事情に基づいてなされたものであり、コンピュータに実装できるメモリ容量に制限がある場合でも、大規模なプログラムの解析を自動的に行うことを可能とするソフトウェア解析装置及びソフトウェア解析方法を提供することを目的とする。

また、本発明の他の目的は、プログラムの解析を行っている際に何らかの要因で解析が中断された場合等に、中断される間隙までに得られた情報を有効に利用することのできるソフトウェア解析装置及びソフトウェア解析方法を提供することである。

【0006】

【課題を解決するための手段】

上記の目的を達成するために、本発明に係るソフトウェア解析装置は、コンピュータプログラムのソースコードを読み込んで、自動的にプログラム解析情報を作成するプログラム解析情報作成手段と、該プログラム解析情報作成手段によって前記プログラム解析情報が作成され次第、適当な単位で分割して所定のデータ記録媒体に格納するプログラム解析情報格納手段と、前記データ記録媒体から前記プログラム解析情報を読み出して、プログラム解析を実行するプログラム解析手段と、を有することを特徴とする。

【0007】

本発明に係るソフトウェア解析方法は、コンピュータプログラムのソースコードを読み込んで、自動的にプログラム解析情報を作成するプログラム解析情報作成工程と、該プログラム解析情報作成工程によって前記プログラム解析情報が作成され次第、適当な単位で分割して所定のデータ記録媒体に格納するプログラム解析情報格納工程と、前記データ記録媒体から前記プログラム解析情報を読み出して、プログラム解析を実行するプログラム解析工程と、を有することを特徴とする。

【0008】

本発明に係るコンピュータ読み取り可能な記録媒体は、コンピュータに、コンピュータプログラムのソースコードを読み込んで、自動的にプログラム解析情報を作成するプログラム解析情報作成機能と、該プログラム解析情報作成機能によって前記プログラム解析情報が作成され次第、適当な単位で分割して所定のデータ記録媒体に格納するプログラム解析情報格納機能と、前記データ記録媒体から前記プログラム解析情報を読み出して、プログラム解析を実行するプログラム解析機能と、を実行させるプログラムを記録したことを特徴とする。

【0009】

【発明の実施の形態】

以下に図面を参照して、本発明の実施の形態について説明する。図1は、本発明の一実施形態のソフトウェア解析装置（以下「本装置」という）における解析情報作成部について説明するための図、図2は本実施形態のソフトウェア解析装置がデータベースに格納する解析情報の構造を示す図、図3は本実施形態のプロ

グラム解析部について説明するための図である。

【0010】

図1において、ソースファイル10は、所定の言語（例えばC言語）で記述されたプログラムのソースコードが記録されたファイルである。プログラム解析情報作成部11は、コンピュータによるバッチ処理によって、読み込まれたソースファイル10のソースコードについて所定の解析処理を行ってプログラム解析情報を作成し、これらをプログラム解析情報格納部12によりデータベース化して、ハードディスク等の情報記録媒体12aに格納する。本実施形態の場合、プログラム解析情報には、図2に示すように、構文解析木（シンボルテーブルを含む）、コールグラフ、フローグラフ、データフロー情報、プログラム依存グラフ、モジュール入出力情報が含まれる。

【0011】

プログラム解析情報作成部11は、図2に示すように、プログラム解析情報を階層化して作成する。このように階層化するのは、上層のプログラム解析情報を作成するときに、下層のプログラム解析情報を利用することによるものである。このとき、一つの層ごとのひとまとまりのプログラム解析情報の作成が終了した段階で、プログラム解析情報格納部12がこれらをトランザクションとして順次データベースに格納する。

【0012】

例えば、コールグラフやフローグラフを作成するには構文解析木が必要となり、データフロー情報を作成するには構文解析木、コールグラフ及びフローグラフが必要となるので、図2の下層のプログラム解析情報から順番にプログラム解析情報格納部12の情報記録媒体12aに格納してゆく。このように、上層のあるプログラム解析情報を作成する場合に、既に作成され、格納されている下層のプログラム解析情報を必要とする関係を、ここでは「階層化」と呼んでいる。

【0013】

仮に、トランザクション機能を用いずに、バッチ処理によってすべてのプログラム解析情報を作成し、それから一括して格納するようにすると、特に大規模な

プログラムの場合には、プログラム解析情報の大きさがワークステーションやパーソナルコンピュータに搭載できるメモリの容量を超えてしまい、それまでに行った処理が無駄になることがある。

【0014】

これに対して、本装置では、プログラム解析情報作成部 11 が作成したプログラム解析情報を、プログラム解析情報格納部 12 が階層ごとに分割し、これらを順次ハードディスク等の情報記録媒体 12a にデータベース化して格納するようにしたことによって、コンピュータ上の限られたメモリ容量の制約から開放されるというメリットがある。

【0015】

また、図 2 のデータ構造の各階層ごとに、それらが作成された段階でトランザクションとしてハードディスク等の情報記録媒体 12a に格納することによって、何らかの障害が発生した場合でも、プログラム解析情報のうち、それまでに格納された部分は安全に保護される。例えば構文解析からプログラム依存グラフの作成まで数日程度を要する大規模なプログラムの解析を行おうとした場合に、途中で何らかの障害が発生してコンピュータがダウンし、メインメモリ上に展開されていたデータが破壊されたとしても、少なくとも既にデータベースに格納された部分については、安全に保護され、次に作業を再開するときは、既にデータベースに格納されているプログラム解析情報については、改めて解析処理を行う必要はなく、それ以降の解析処理から再開すればよいので、重複した作業を回避することができる。

【0016】

なお、本実施形態では、データベースとしては、オブジェクト指向データベースソフトウェアを利用し、プログラム解析情報は、オブジェクト指向言語、例えば C++ などを用いて作成する。このようなデータベースとするのは、プログラム解析情報は複雑なものであるため、リレーショナルデータベースを用いたのでは格納が困難なためである。また、作成されたプログラム解析情報を、作成された構造のまま格納することができるので、便利である。

【0017】

バッチ解析部 13 は、プログラム解析情報格納部 12 によって情報記録媒体 12 a に格納された情報を用いて、バッチ処理で解析することができる解析処理を行い、その結果をバッチ解析結果ファイル 14 に格納する。このバッチ処理中は、オペレータからの指示などは特に必要なく、本装置が自動的に解析を行う。

バッチ解析結果ファイル 14 には、例えばプログラムの規模に関連するメトリクス (metrics) 情報や冗長度情報、保守用文書情報などが含まれる。「メトリクス情報」とは、ソフトウェアの数値化指標に関する情報であり、本装置では、量的複雑さを表すメトリクスと、質的複雑さを表すメトリクスを計測し、その情報を作成する。量的複雑さのメトリクスとしては、プログラムの物理的な記述量を計測するサイズ・メトリクスと、制御構造の複雑さを計測するサイクロマティック数の二種類を計測し、質的複雑さのメトリクスとしては、モジュールの内容そのものを表す凝集度と結合度を計測する。「冗長度情報」とは、単一の手続きについて、出力に影響しない冗長な文に関する情報である。冗長な文は出力に影響しないので、ソースコードから削除しても外部に対する出力に変化はない。冗長度情報に従って、このような冗長な文を確認して対応策をとっておくことによって、意図しない動作の発生等を防止することができる。「保守用文書情報」とは、プログラムを保守する際に用いるドキュメント類で、定義されている手続きや型、変数名などの一覧や、それぞれの手続き等がどこに関連したものであるかを記載したものであり、例えば、他人が作成したプログラムを理解するときに使われる。「保守用文書情報」を取得しておくことで、ハイパーテキスト方式で情報提供することができ、紙ベースのドキュメントと比べると、格段に便利になる。

【0018】

冗長度情報、メトリクス情報及び保守用文書情報などは、バッチ解析処理の結果生成されるが、バッチ解析処理の結果生成されるものは、これらに限られず、必要に応じて種々のバッチ解析処理を行い、その結果、それぞれのバッチ解析処理に対応した種々の情報を生成するよう、バッチ解析部 13 を設計することが可能である。

【0019】

図2において、「シンボルテーブル」とは、変数の情報の表である。例えば、同じ「a」という文字を使っている場合でも、それらが同じ変数を指す場合と、異なる変数を指す場合がある。例えば、異なる二つの関数の中で、その変数が大域変数であれば、同じものであるし、それぞれの関数においてローカルに定義された変数であれば、別々の変数となる。したがって、同じ文字で記述された変数が複数あったときに、それらが同じものなのか異なるものなのかは、直ちには識別できない。このようなシンボルに関する情報を管理するのがシンボルテーブルであり、このテーブルを参照することによって、同じシンボルで表されたものが同じものか異なるものかを知ることができる。

【0020】

図2において、「コールグラフ」とは、手続き（C言語では「関数」）の間の呼び出し関係を示した情報である。このコールグラフに基づいて、図3のプログラム解析部20₁がこれをノードとエッジを用いてグラフィック表示させた場合、グラフィック表示されたグラフでは、ノードが手続きを表し、エッジ（矢印）が手続き間の関係を表す。なお、コールグラフをグラフィック表示させると、手続き間の呼び出し関係が視覚的に把握できるだけでなく、プログラムの構造化の程度や内容の理解や、望ましくない手続き呼び出しの検出が容易になる。

【0021】

図2において、「フローグラフ」とは、ある一つの手続き内の制御フローに関する情報である。フローグラフに基づいて、プログラム解析部20₂がこれを、ノードとエッジを用いてグラフィック表示させると、ノードは基本ブロック（連続した文の列からなり、制御が先頭の文に移された後途中で停止したり分岐したりせず、最後の文で制御が離れるものをいう）を表し、エッジは基本ブロック間の制御のフローを表す。

【0022】

フローグラフについても、プログラム解析部20₂によって、その情報をグラフィック表示させると、その手続き内の制御フローが視覚的に把握できるだけでなく、プログラム中のある場所（一つの文か、または一つの変数）に影響を与える部分もしくは与えられる部分の解析や、望ましくない制御フローの検出を可能

にする。例えば、ある場所が制御依存する部分はその場所から入口ノードに向かってフローグラフを辿ることで検出される。また、制御が到達することのないデッドコードは、入口ノード以外のノードであって、流入するエッジのないノードとして表現される。

【0023】

図2において、「データフロー情報」とは、例えばある場所で定義された変数がどこで使用されるかといったことに関するデータの流れを解析した結果の情報である。プログラム解析部による対話処理でデータフロー異常検査を行うときは、図2に階層的に示したプログラム解析情報のうち、構文解析木、コールグラフ、フローグラフ、そしてデータフロー情報までが必要となる。データフロー異常とは、データに対する不当なイベントの組み合わせである。すべてのデータは、定義した上で使用し、最後に未定義にするという一連の順序を守って使用しなければならないが、このような順序を守らない不当な組み合わせは、データフロー異常となる。

【0024】

データフロー異常が存在しても、ソースコードをコンパイルすることは可能である。しかし、実際にコンパイルしたプログラムを実行させているときに、異常なパスを通るとデータフロー異常が顕在化することがある。したがって、このような異常は、事前にソースコードの段階で除去しておくのが望ましい。データフロー異常の検出については、本出願人による特許出願、特願平9-32415及び特願平9-32452を参照することができる。なお、ここでは、データフロー異常検査を対話処理で行う場合について説明したが、データフロー異常検査は、バッチ処理で行うこともできる。

【0025】

図2において、「プログラム依存グラフ」とは、制御の依存関係と、データの依存関係をグラフで表現したものである。プログラム依存グラフが作成されると、プログラムの影響解析を実行することができる。影響解析とは、プログラム中のある場所（一つの文か、または一つの変数）を変更すると、それがどこまで影響を与えるか、あるいは影響を与えないかを求める解析である。

【0026】

次に、図3を参照して、プログラム解析部について説明する。同図には、プログラム解析部 20_1 、 \dots 、 20_i が示されている。プログラム解析部 20_1 等は、プログラム解析情報格納部12によってハードディスク装置等にデータベース化して格納されているプログラム解析情報に基づいて、種々のプログラム解析を実行する部分である。各プログラム解析部 20_1 、 \dots 、 20_i は解析する種類に対応して設けられている。プログラム解析部 20_1 等は、プログラム解析情報格納部12から必要なプログラム解析情報を取り出してきて、所定の解析処理を行い、その解析結果 21_1 等をコンピュータ画面上にGUI (Graphical User Interface) 22_1 等により表示させる。オペレータは、この結果を見て、何らかの指示が必要な場合には、GUI 22_1 等を介してプログラム解析部 20_1 等へ指示を送り、プログラム解析部 20_1 等に再度処理を行わせ、その結果を表示させる。このような処理を繰り返しながら、プログラム解析を行ってゆく。このように、オペレータがGUI 22_1 等を介してコンピュータに指示し、それに対してコンピュータが処理を行うような操作を対話処理と呼ぶ。

【0027】

本実施形態では、プログラム解析部 20_1 は、コンピュータ画面にコールグラフをグラフィック表示させる解析処理を行う。これにより、コールグラフの内容を実際に視覚的に把握できるように、プログラム内の関数の呼出し関係の理解が容易となる。この場合、プログラム解析部 20_1 は、プログラム解析情報として、構文解析木及びコールグラフをプログラム解析情報格納部12から読み出して来る。また、プログラム解析部 20_2 は、コンピュータ画面にフローグラフをグラフィック表示させる解析処理を行う。これにより、フローグラフの内容を実際に視覚的に把握できるように、ある関数の内部における変数の流れの理解が容易となる。この場合、プログラム解析部 20_2 は、プログラム解析情報として、構文解析木及びフローグラフをプログラム解析情報格納部12から読み出して来る。

【0028】

プログラム解析部が実行する解析処理の他の例として、影響範囲解析がある。これは、プログラム解析情報格納部 12 によりデータベースからプログラム解析情報として、構文解析木、コールグラフ、フローグラフ、プログラム依存グラフ、モジュール入出力情報を読み出して、プログラムのソースコードのある行が、他のどの行まで影響を及ぼしているかを解析する。具体的には、プログラムのソースコードを表示し、そこである行を指定して、その行がソースコードのどこに影響しているかを、例えば色を変えて表示する。そして、その影響されている行が更にどの行に影響しているか、ということを順次調べる、ということを行う。こういう対話的な処理を繰り返すことによって、オペレータは、例えば他人が作成したプログラムの構造を利用しやすくなる。

【0029】

更に、プログラム解析部が行う別の例としては、構造解析、データフロー異常解析などがある。

図3において、GUI 23 は、バッチ解析結果ファイル 13 の内容を、オペレータが見ることができるように表示させるためのものである。また、GUI 連動部 24 は、GUI 22₁, ..., 22_i 及び 23 のうちで任意の数の GUI をコンピュータ画面に表示させ、例えばある GUI 画面に表示されたソースプログラムのある行を指定すると、他の GUI 画面に表示されたフローグラフの対応する部分が色違いで表示されるというように、複数の GUI 画面を関連付けて表示するための部分である。これを用いることによって、プログラム内容の理解がより容易となる。

【0030】

本装置では、プログラム解析情報作成部 11 において、どの解析情報までを作成するかを、オペレータが任意に指定できるようにしてもよい。このようにすると、例えば、コールグラフを表示するだけという場合には、それより上位の解析情報は必要ないので、その旨を指定しておくことにより、バッチ処理による解析処理がより高速化する。

【0031】

次に、本発明の記録媒体について説明する。本発明のソフトウェア解析装置は

、プログラムを読み込んだコンピュータが、そのプログラムに従って動作することによって実現される。したがって、このようプログラムを記録した記録媒体、例えばフロッピーディスク、CD-ROM、光磁気ディスク、磁気テープ、半導体記憶装置などをコンピュータに装着し、当該プログラムを読み込ませることによって、コンピュータ上で本発明を実施することが可能となる。

【0032】

尚、本発明は、上記実施形態に限定されるものではなく、その要旨の範囲内で種々の変更が可能である。

【0033】

【発明の効果】

以上説明したように、本発明によれば、プログラム解析情報作成手段によって作成されたプログラム解析情報を複数に分割し、それぞれを順次所定のデータ記録媒体に格納するので、大規模なプログラムに対してプログラム解析情報を得ようとした場合に、コンピュータに実装できるメモリ容量に制限があっても、大規模なプログラムのプログラム解析情報を安全に得ることができる。

【図面の簡単な説明】

【図1】

本発明の一実施形態のソフトウェア解析装置における解析情報作成部について説明するためのである。

【図2】

本発明の一実施形態のソフトウェア解析装置がデータベースに格納するプログラム解析情報の構造を示した図である。

【図3】

本発明の一実施形態のプログラム解析部について説明するための図である。

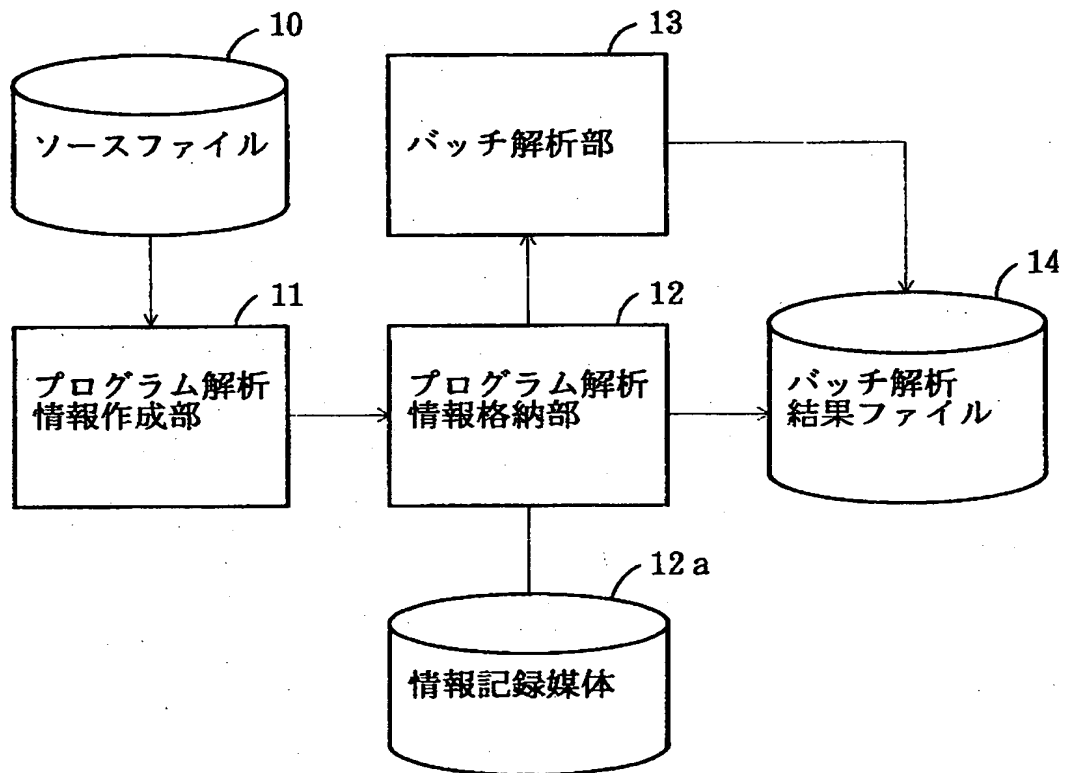
【符号の説明】

- 10 ソースファイル
- 11 プログラム解析情報作成部
- 12 プログラム解析情報格納部
- 13 バッチ解析部

1.4 バッチ解析結果ファイル
20₁, ... 20_i プログラム解析部
21₁, ... 21_i 解析結果
22₁, ... 22_i, 23 GUI
24 GUI連動部

【書類名】 図面

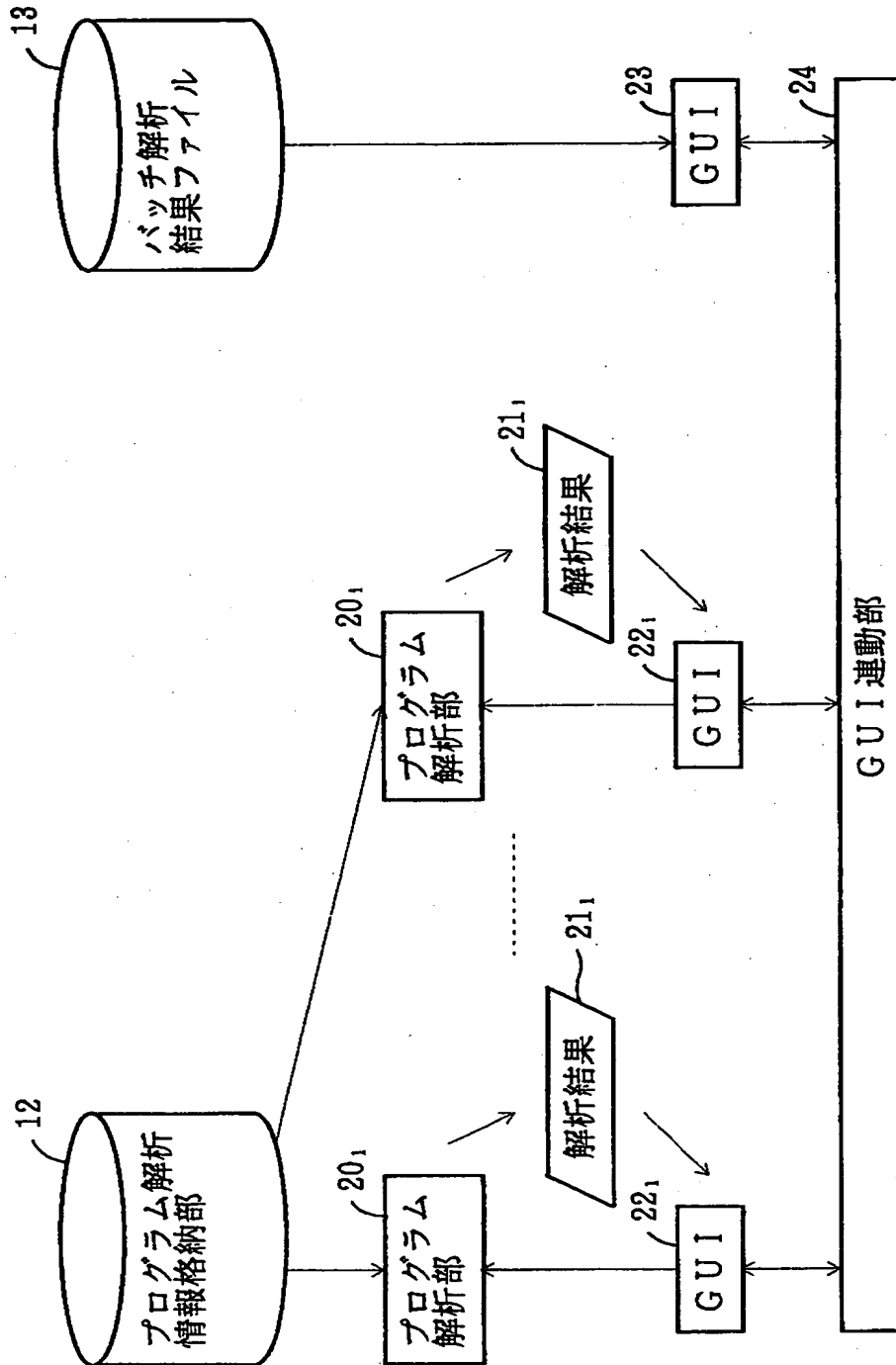
【図1】



【図 2】

メトリクス情報	冗長度情報	保守用文書情報	-----
プログラム依存グラフ		モジュール入出力情報	
データフロー情報			
コールグラフ		フローグラフ	
構文解析木（シンボルテーブルを含む）			

【図3】



【書類名】 要約書

【要約】

【課題】 コンピュータに実装できるメモリ容量に制限がある場合でも、大規模なプログラムの解析を自動的に行うことを可能とするソフトウェアシステム解析装置及びソフトウェア解析方法を提供する。

【解決手段】 プログラム解析情報格納部 12 は、プログラム解析情報作成部 11 が、ソースファイル 10 に基づいて作成したプログラム解析情報をハードディスク等の不揮発性の情報記録媒体にデータベース化して格納する。このとき、ひとまとまりの解析情報が作成され次第、これらをトランザクションとして順次データベースに格納する。これにより、コンピュータ上の限られたメモリ容量の制約から開放され、また、解析の途中で何らかの障害が発生した場合でも、既にデータベースに格納されたプログラム解析情報は安全に保護され、後に解析を再開するときに、未格納の部分から解析処理を行えばよい。

【選択図】 図 1

【書類名】

職権訂正データ

【訂正書類】

特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】

000006655

【住所又は居所】

東京都千代田区大手町2丁目6番3号

【氏名又は名称】

新日本製鐵株式会社

【代理人】

申請人

【識別番号】

100091269

【住所又は居所】

東京都杉並区和田2丁目3番8号 半田国際特許事務所

【氏名又は名称】

半田 昌男

出 願 人 履 歴 情 報

識別番号 [000006655]

1. 変更年月日	1990年 8月10日
[変更理由]	新規登録
住 所	東京都千代田区大手町2丁目6番3号
氏 名	新日本製鐵株式会社